

---

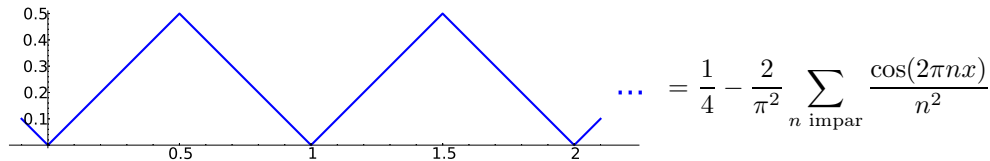
## ANÁLISIS DE FOURIER

---

### Motivación e historia

La idea básica del análisis de Fourier es que una *señal* se descomponer en *tonos puros*. Por supuesto, hay que precisar mucho para llegar a enunciados matemáticos de esta afirmación. Para fijar ideas, pensemos que una señal es una función con cierta regularidad, y quizá propiedades de periodicidad, y los tonos puros son funciones que de alguna forma tienen frecuencia fija que toma valores en cierto conjunto, por ejemplo  $\sin(\alpha nx)$  y  $\cos(\alpha nx)$  con  $n \in \mathbb{Z}$ . Tomando una terminología física, a veces los tonos puros se llaman *armónicos*, y por ello *análisis armónico* es sinónimo de análisis de Fourier.

Simplemente por dar una imagen previa del tipo de expresiones que aparecen en el análisis de Fourier, mencionamos como ejemplo la descomposición de una onda triangular como superposición de cosenos de frecuencias enteras, que son los tonos puros o armónicos en este caso:



La idea tiene su origen en el siglo XVIII, con los trabajos de L. Euler y J. d'Alembert acerca de la ecuación de ondas, la cual admite una solución con un aspecto simétrico para datos expresables como una superposición de senos y cosenos de frecuencias enteras. Ninguno de los dos pensaba que esto se aplicase a funciones 1-periódicas generales (con cierta regularidad), mientras que D. Bernoulli defendía la opinión contraria.

La famosa memoria de J. Fourier sobre la propagación del calor [Fou88], publicada en 1822, no resolvió esta cuestión, ni siquiera con los niveles de rigor matemático de entonces (por ejemplo, Fourier resolvía ecuaciones lineales de infinitas variables trabajando con determinantes). Sin embargo dio buenas razones para confiar que las funciones 1-periódicas son superposición de tonos puros de la forma  $A \cos(2\pi nx)$ ,  $B \sin(2\pi nx)$  con  $n \in \mathbb{Z}$ . Por ello tiene cierta lógica, que hablemos actualmente de *análisis de Fourier*, a pesar de la contribución fundamental de muchos matemáticos. La primera prueba de la convergencia de los desarrollos de Fourier, bajo condiciones adecuadas, se debe a P.G.L. Dirichlet en 1829.

Muchas de las aplicaciones del análisis de Fourier se basan en la conveniencia de eliminar o modificar ciertos rangos de las frecuencias que componen una función porque no tienen influencia significativa en el resultado o porque son ruido. La reducción de rangos de frecuencias se traduce a menudo en la facultad de poder comprimir una señal de audio o vídeo.

Dentro de las Matemáticas y la Física, una utilidad básica es que hay problemas que son difíciles de entender para funciones generales pero son más sencillos para tonos puros. El análisis de Fourier tiene incluso ciertas implicaciones en la Filosofía de la Ciencia, pues participa en el propio concepto de experimento físico a través de la *interpretación de Copenhague* de la mecánica cuántica (la imperante hoy en día, aunque físicos de prestigio como Einstein nunca la aceptaron). Sin entrar en detalles, según ella, los sistemas tienen funciones de onda y un experimento físico detecta los tonos puros que las componen con una probabilidad que depende de su amplitud.

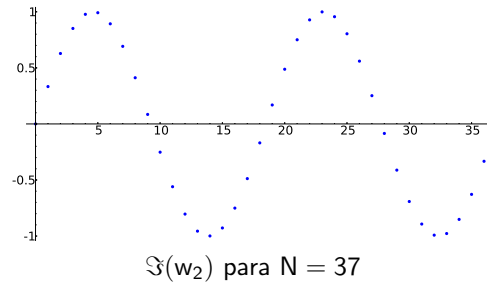
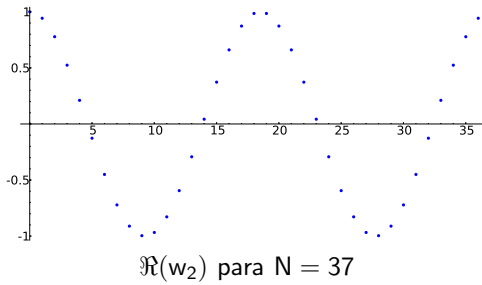
### Desarrollos de Fourier

En lo que sigue, vamos a ver diferentes formas de análisis de Fourier. La primera es la más sencilla pero también la más relevante en las aplicaciones actuales que tratan señales digitalizadas.

Si *muestreamos* una señal compleja tomando  $N$  valores por cada periodo, el resultado se puede representar como una función  $f : \mathbb{Z}_N \rightarrow \mathbb{C}$  donde  $\mathbb{Z}_N$  son las clases de enteros módulo  $N$ . Con esto se quiere representar que el valor de  $f(k)$  es el mismo que el de  $f(k+N)$ . Lo que queremos hacer es expresar  $f$  como una superposición de tonos puros discretos, concretamente de

$$(1) \quad w_m(n) = e\left(\frac{mn}{N}\right) \quad \text{donde} \quad e(x) = e^{2\pi ix}.$$

La función  $e(x)$  es sólo una manera cómoda de encapsular el seno y el coseno. No tiene mayor propósito que simplificar las fórmulas y hacerlas más simétricas. Dibujar las partes real e imaginaria de  $w_m$  para  $m$  fijo y  $N$  grande, justifica que consideremos estas funciones como tonos puros.



La siguiente igualdad se reduce a la suma de una serie geométrica:

$$(2) \quad \frac{1}{N} \sum_{m \in \mathbb{Z}_N} w_m(n) = \delta(n) \quad \text{donde} \quad \delta(n) = \begin{cases} 1 & \text{si } n \equiv 0 \pmod{N} \\ 0 & \text{si } n \not\equiv 0 \pmod{N} \end{cases}$$

Si sustituimos esta relación en la fórmula trivial  $f(n) = \sum_{m \in \mathbb{Z}_N} f(m)\delta(n-m)$ , tras invertir el orden de sumación, se concluye el resultado deseado para cualquier  $f : \mathbb{Z}_N \rightarrow \mathbb{C}$

$$(3) \quad \boxed{f(n) = \frac{1}{N} \sum_{m \in \mathbb{Z}_N} \hat{f}(m)e(nm/N) \quad \text{donde} \quad \hat{f}(n) = \sum_{m \in \mathbb{Z}_N} f(m)e(-nm/N).}$$

Esta expresión de  $f$  en términos de  $w_m$  se llama *desarrollo de Fourier* y a la función  $n \mapsto \hat{f}(n)$  se le llama *transformada de Fourier discreta*, más conocida por sus siglas en inglés, DFT. Cada  $\hat{f}(n)$  se dice que es un *coeficiente de Fourier*.

Si están almacenadas en memoria las raíces  $N$ -ésimas de la unidad, cada  $\hat{f}(n)$  requiere  $N$  multiplicaciones, y por tanto hallar todos ellos requeriría  $N^2$ . Sin embargo, no es difícil percatarse de que los cálculos se reducen enormemente usando la periodicidad de las funciones trigonométricas. Al refinar esta idea, se obtiene un algoritmo llamado *transformada rápida de Fourier* y abreviada habitualmente como FFT. Normalmente se cita [CT65] como el artículo con el que nació la FFT en 1965, sin embargo, algunos autores [HJB85] han señalado que el método se remonta a un trabajo póstumo de C.F. Gauss.

El algoritmo permite pasar de  $N^2$  operaciones a una cantidad del orden de  $N \log N$ . Una reducción tan drástica tiene una influencia fundamental en las aplicaciones (en [Str93] se dice que es el método numérico más importante de nuestros días). Según una opinión recogida en [Ger99], es tan sumamente necesaria, que se decidió que la FFT fuera de dominio público y alejarla de las patentes.

Si  $N$  es grande, la nube de puntos que compone las ondas discretas  $w_m$  de (1), se parecerán más a senos y cosenos, que convenientemente escalados tendrán frecuencias enteras. Para materializar esta idea, escribamos

$x = n/N$ , si  $F : \mathbb{Z}_N \rightarrow \mathbb{C}$  entonces  $f(x) = F(Nx) = F(n)$  cumple  $f(x+1) = f(x)$ . De acuerdo con (3), se tiene

$$(4) \quad f(x) = \sum_{m \in \mathbb{Z}_N} \frac{\widehat{F}(m)}{N} e(mx) \quad \text{donde} \quad \frac{\widehat{F}(m)}{N} = \frac{1}{N} \sum_{k \in \mathbb{Z}_N} F(k/N) e(-mk/N).$$

La última expresión es una suma de Riemann que tenderá a la integral si  $N$  es muy grande, en cuyo caso  $x = n/N$  se comportará como si recorriera todo  $[0, 1]$ . Esto sugiere que si  $f : \mathbb{R} \rightarrow \mathbb{C}$  es 1-periódica, lo cual a veces se abrevia como  $f : \mathbb{T} \rightarrow \mathbb{C}$ , se tiene el siguiente *desarrollo de Fourier*

$$(5) \quad \boxed{f(x) = \sum_{n=-\infty}^{\infty} \widehat{f}(n) e(nx) \quad \text{donde} \quad \widehat{f}(n) = \int_0^1 f(x) e(-nx) dx.}$$

Por supuesto, el argumento está lejos de ser una demostración y son necesarias condiciones de regularidad. Por ejemplo, si  $f \in C^2(\mathbb{T})$ , (donde, de nuevo,  $\mathbb{T}$  indica la 1-periodicidad), la fórmula es correcta con convergencia absoluta con cierta rapidez. Ya si  $f \in C^1(\mathbb{T})$  la convergencia está asegurada y es uniforme (aunque quizá no absoluta). Uno de los grandes hitos del análisis de Fourier es el difícilísimo *teorema de Carleson-Hunt* que asegura que la igualdad en (5) es cierta en casi todo punto para una función que esté en cualquier espacio  $L^p(\mathbb{T})$  con  $p > 1$ .

Obviamente no podemos descomponer una función que no sea 1-periódica como suma de senos y cosenos de frecuencias enteras, a través de  $e(nx)$ . Sin embargo todavía hay una oportunidad de lograrlo cuando las frecuencias no son enteras. El truco es pensar fantasiosamente que una función que no es periódica es como si fuera periódica con periodo infinitamente grande.

Una función  $T$ -periódica  $f$  se escribe como  $f(x) = g(x/T)$  donde  $g$  es 1-periódica. Utilizando (5) para  $g$  y sustituyendo  $x$  por  $x/T$  se tiene

$$(6) \quad f(x) = \sum_{n=-\infty}^{\infty} \widehat{g}(n) e\left(\frac{n}{T}x\right) \quad \text{con} \quad \widehat{g}(n) = \int_{-1/2}^{1/2} g(x) e(-nx) dx.$$

Con un cambio de variable e incorporando el coeficiente de Fourier en el sumatorio:

$$(7) \quad f(x) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-T/2}^{T/2} f(x) e\left(-\frac{n}{T}x\right) dx \cdot e\left(\frac{n}{T}x\right)$$

Si  $T$  es muy grande,  $n/T$  recorrerá puntos cercanos de  $\mathbb{R}$ . Procediendo como antes, pensando en la suma como una suma de Riemann, se intuye que

$$(8) \quad \boxed{f(x) = \int_{-\infty}^{\infty} \widehat{f}(\xi) e(\xi x) d\xi \quad \text{donde} \quad \widehat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e(-\xi x) dx.}$$

A esta fórmula se le llama *fórmula de inversión* y a  $\widehat{f}$  se le llama *transformada de Fourier* de  $f$  y fue introducida por el propio Fourier en [Fou88]. De nuevo, no hemos probado (8), sólo hemos dado indicios para que sea correcta (como hizo Fourier). La no acotación de  $\mathbb{R}$ , limita la certeza de (8) a funciones tales que tengan cierta regularidad pero también cierto decaimiento. Por ejemplo, es válida para funciones  $f \in C^2$  tales que  $|f^{(k)}(x)| |x|^{-1-\epsilon} \rightarrow 0$ ,  $k = 0, 1, 2$  cuando  $x \rightarrow \infty$ .

### Proyecciones, convoluciones y filtros

Vistas las similitudes entre (3), (5) y (8), cabe preguntarse si hay una teoría general de la que se deduzcan estas fórmulas como casos particulares. La respuesta es afirmativa y existe un desarrollo de Fourier asociado a

cada grupo abeliano localmente compacto [Kat76]. Las fórmulas anteriores corresponden, respectivamente, a los grupos  $(\mathbb{Z}_N, +)$ ,  $(\mathbb{R}/\mathbb{Z}, +)$  y  $(\mathbb{R}, +)$ .

Sin entrar en este tipo de teorías, es posible desarrollar cierta intuición considerando la descomposición (3) en el contexto de espacios vectoriales euclídeos lo cual, por extensión también arrojan luz sobre (5) y (8).

Recordemos que en un espacio vectorial (complejo) de dimensión finita, si  $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_n\}$  es una base ortonormal, cualquier vector  $\vec{x}$  se escribe como

$$(9) \quad \vec{x} = a_1\vec{u}_1 + a_2\vec{u}_2 + \dots + a_n\vec{u}_n \quad \text{con} \quad a_i = \langle \vec{u}_i, \vec{x} \rangle.$$

Si tenemos un subespacio con una base ortonormal  $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k\}$ , la proyección (ortogonal) sobre él viene dada por una fórmula similar:

$$(10) \quad \text{Pr}(\vec{x}) = \sum_{m=1}^k \langle \vec{u}_m, \vec{x} \rangle \vec{u}_m.$$

Las funciones  $\{f : \mathbb{Z}_N \rightarrow \mathbb{C}\}$  forman un espacio vectorial compleja y si introducimos el producto escalar

$$(11) \quad \langle f, g \rangle = \frac{1}{N} \sum_{n \in \mathbb{Z}_N} \bar{f}(n)g(n)$$

entonces las funciones  $\{w_m\}_{m=0}^{N-1}$  definidas en (1) forman una base ortonormal gracias a (2). Con ello, el desarrollo de Fourier discreto (3) no es más que (9).

En el caso de (5) es natural considerar el espacio vectorial complejo completo (un *espacio de Hilbert*) dado por  $L^2(\mathbb{T}) = \{f : \mathbb{R} \rightarrow \mathbb{C} \text{ 1-periódica, } f \in L^2[0, 1]\}$  con el producto escalar  $\langle f, g \rangle = \int_0^1 \bar{f}(x)g(x) dx$ . Las funciones  $\{w_n\}_{n \in \mathbb{Z}}$  con  $w_n(x) = e(n x)$  son ortonormales pero en el sentido estricto del álgebra lineal (que no trata cuestiones de convergencia) no se puede decir que es una base. Sin embargo, la comparación entre (5) y (9) nos lleva a pensar que en cierto modo sí lo es y que (5) no es otra cosa que la expresión en esta base. En rigor, se dice que  $\{w_n\}_{n \in \mathbb{Z}}$  forman un *sistema ortonormal completo* del espacio de Hilbert  $L^2(\mathbb{T})$ .

En el caso de (8) el producto escalar natural es  $\langle f, g \rangle = \int_{-\infty}^{\infty} \bar{f}(x)g(x) dx$  en  $L^2(\mathbb{R})$  pero parece que ahora deberíamos considerar un conjunto no numerable  $\{w_\xi\}_{\xi \in \mathbb{R}}$  que se sale del espacio y por tanto su ortogonalidad es difícilmente interpretable. Sin embargo, todo funciona como si así fuera.

Pensemos por ejemplo, en el teorema de Pitágoras aplicado a (9), que nos da  $\|\vec{x}\|^2 = \sum_{m=1}^n |\langle \vec{u}_m, \vec{x} \rangle|^2$ . En cada una de las tres situaciones discutidas anteriormente, nos da respectivamente las siguientes fórmulas que no son en absoluto triviales:

$$(12) \quad \sum_{n \in \mathbb{Z}_N} |f(n)|^2 = \frac{1}{N^2} \sum_{n \in \mathbb{Z}_N} |\hat{f}(n)|^2, \quad \int_0^1 |f|^2 = \sum_{n \in \mathbb{Z}} |\hat{f}(n)|^2, \quad \int_{-\infty}^{\infty} |f|^2 = \int_{-\infty}^{\infty} |\hat{f}|^2.$$

Por ejemplo, si tomamos la función 1-periódica que coincide con  $f(x) = x$  en  $[0, 1)$  entonces  $\hat{f}(0) = 1/4$  y  $\hat{f}(n) = i/(2\pi n)$  para  $n \neq 0$ . La segunda fórmula da una identidad que equivale a  $\pi^2/6 = \sum_{n=1}^{\infty} n^{-2}$ .

La idea de la proyección (10) es muy natural en las aplicaciones del análisis de Fourier. Por ejemplo, una señal de audio 1-periódica, en principio requiere infinitos números complejos para ser representada a través de sus coeficientes de Fourier (5). Sin embargo el espectro audible está entre 20Hz y 20kHz, con lo cual no perdemos nada si proyectamos sobre el subespacio que corresponde a estas combinaciones de frecuencias, lo que corresponde a

$$(13) \quad f \mapsto \sum_{20 \leq |n| \leq 20000} \hat{f}(n)e(nx).$$

En otras ocasiones, se quieren exagerar algunas frecuencias o atenuarlas sin perderlas del todo (por ejemplo, aumentar los graves o los agudos). Matemáticamente la operación que generaliza la proyección en el caso de funciones 1-periódicas sería multiplicar cada coeficiente por un peso,

$$(14) \quad f \mapsto \sum_{n \in \mathbb{Z}} w(n) \widehat{f}(n) e(nx).$$

De cara a las aplicaciones, es interesante saber a qué operación sobre la función corresponde esta acción sobre los coeficientes de Fourier. La respuesta, está relacionada con la *convolución*, que en cada uno de los casos antes considerados,  $f : \mathbb{Z}_N \rightarrow \mathbb{C}$ ,  $f : \mathbb{T} \rightarrow \mathbb{C}$  y  $f : \mathbb{R} \rightarrow \mathbb{C}$ , se define respectivamente como

$$(15) \quad (f * g)(n) = \sum_{m \in \mathbb{Z}_N} f(n-m)g(m), \quad (f * g)(x) = \int_0^1 f(x-t)g(t) dt, \quad (f * g)(x) = \int_{-\infty}^{\infty} f(x-t)g(t) dt.$$

La convolución es una operación conmutativa:  $f * g = g * f$ .

El resultado fundamental sobre convoluciones en el ámbito del análisis de Fourier es que en cualquiera de las tres situaciones:

$$(16) \quad \widehat{f * g} = \widehat{f} \widehat{g}$$

Es decir, la multiplicación de coeficientes o transformadas de Fourier, como en (14), corresponde a una convolución de la función.

La prueba es similar en los tres casos. Consideremos el tercero:

$$(17) \quad \widehat{f * g}(\xi) = \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} f(x-t)g(t) dt \right) e(-x\xi) dx = \int_{-\infty}^{\infty} g(t) \left( \int_{-\infty}^{\infty} f(x-t)e(-x\xi) dx \right) dt.$$

Con el cambio  $x = y + t$  en la integral interior, se tiene

$$(18) \quad \widehat{f * g}(\xi) = \int_{-\infty}^{\infty} g(t) e(-\xi t) \left( \int_{-\infty}^{\infty} f(y) e(-y\xi) dy \right) dt = \widehat{g}(\xi) \widehat{f}(\xi).$$

Acciones como (14) o sus análogos en la situación finita o no periódica, se pueden considerar *filtros*. Un valor nulo de  $w$  corresponde a una frecuencia que desaparece totalmente y otros valores, corresponde a dar mayor o menor importancia a una frecuencia. Lo que hemos probado a través de (16) es que estos filtros siempre se pueden entender como convoluciones.

### Simetrías y decaimiento de los coeficientes de Fourier

Está claro que desde el punto de vista práctico, no podemos guardar todo el desarrollo de Fourier de una función 1-periódica arbitraria en una máquina. En principio sí podríamos hacerlo con los desarrollos de Fourier en  $\mathbb{Z}_N$  porque son finitos pero en ambos casos, con vistas a la compresión, sólo desearíamos guardar la información significativa.

Para fijar ideas, centrémonos en el caso de las funciones  $f : \mathbb{T} \rightarrow \mathbb{C}$ . Un hecho fundamental es que la regularidad implica decaimiento de los coeficientes de Fourier y cuando tenemos coeficientes de Fourier que decaen rápidamente y basta con almacenar los primeros de ellos. Concretamente, para una función  $f \in C^k(\mathbb{T})$  se cumple

$$(19) \quad |\widehat{f}(n)| \leq K |n|^{-k} \quad \text{para cierta constante } K \text{ (dependiente de } f \text{ y } k).$$

La prueba se reduce a una integración sucesiva por partes, usando la periodicidad para eliminar los términos de frontera:

$$(20) \quad \widehat{f}(n) = \int_0^1 f(x) e(-nx) dx = \frac{1}{2\pi i n} \int_0^1 f'(x) e(-nx) dx = \dots = \frac{1}{(2\pi i n)^k} \int_0^1 f^{(k)}(x) e(-nx) dx$$

donde se ve que  $K$  está controlada por la norma 1 de  $f^{(k)}$  y por tanto el mismo resultado se consigue con tal de que  $f^{(k)}$  sea integrable, incluso si no está definida en un conjunto de medida cero, como ocurre con singularidades como en la de  $f(x) = |x|$ .

El problema práctico es que si tomamos un fragmento de una señal, típicamente cuando la hacemos periódica el resultado es discontinuo en  $\mathbb{T}$  y la baja regularidad impide la convergencia rápida de la serie de Fourier. Nuestro objetivo es el caso finito pero para entender la idea, veamos un método sencillo en el caso 1-periódico para forzar una mínima regularidad. Este método depende de la relación entre las simetrías y el desarrollo de Fourier.

Si  $f : \mathbb{T} \rightarrow \mathbb{R}$  es par  $\hat{f}(n) = \int_{-1/2}^{1/2} f(x)e(-nx) dx$  será una real, pues la parte imaginaria de  $e(-nx)$  es impar. El desarrollo de Fourier (5) se puede reducir en este caso a una serie de cosenos. Análogamente, si  $f$  es impar, entonces  $\hat{f}(n)$  será imaginario puro y la serie de Fourier se puede expresar como una serie de senos.

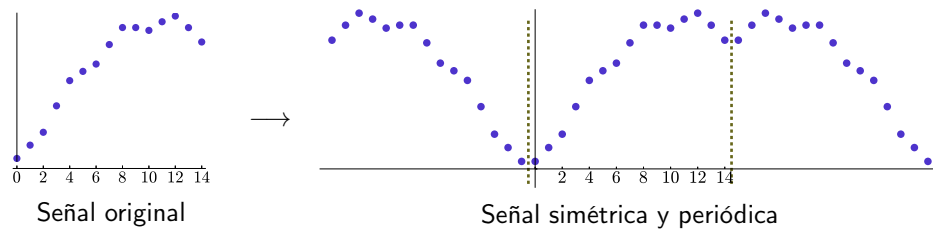
Esta expresión en serie de cosenos o senos dependiendo de la simetría de la función se extiende al caso complejo sin más que considerar partes reales e imaginarias.

$$(21) \quad f_{\text{par}}(x) = \sum_{-\infty}^{\infty} a_n \cos(2\pi nx), \quad f_{\text{impar}}(x) = \sum_{-\infty}^{\infty} b_n \sin(2\pi nx).$$

Volviendo a la regularidad, una función  $f : [0, 1] \rightarrow \mathbb{C}$  típicamente se vuelve discontinua al hacerla 1-periódica, incluso si  $f \in C^\infty[0, 1]$ . Una forma de evitarlo es considerar la función 1-periódica  $g$  que coincide con  $f(2x)$  en  $[0, 1/2]$  y que vale  $f(-2x)$  para  $x \in [-1/2, 0]$ . La función  $g$  tiene toda la información contenida en  $f$  porque  $f(x) = g(x/2)$  en  $[0, 1]$  y además, si  $f \in C^1[0, 1]$ ,  $g$  es continua y  $C^1$  en cualquier intervalo que no contenga enteros. Según (19) esto ya asegura que sus coeficientes decaen al menos tan rápido como  $n^{-1}$ .

En el caso discreto ocurre algo similar. La falta de “regularidad”, es decir, las variaciones abruptas, locales hacen que  $\hat{f}$  no tenga un “decaimiento”, es decir, que ciertos valores dominen sobre el resto, lo cual impide podamos comprimir la señal sin muchas pérdidas.

Adaptando la solución anterior, para evitar este problema se sustituye la señal por otra de periodo doble obtenida por simetría a través de  $-1/2$  (o equivalentemente de  $N - 1/2$ ).



La simetría por  $n = -1/2$  sugiere que estamos trasladando los tonos puros, y como duplicamos el periodo, deberían pasar a ser  $\tilde{w}_m(n) = e(m(n + 1/2)/2N)$ , en lugar de (1). La simetría provoca que el desarrollo de Fourier se convierta en un desarrollo en cosenos de la forma  $\cos(\pi(m + 1/2)n/N)$ . Por las razones antes indicadas y porque en ella sólo participan números reales (si la señal es real), la nueva transformada sustituye en muchas aplicaciones a la transformada de Fourier discreta.

Dada  $f : \{0, 1, \dots, N - 1\} \rightarrow \mathbb{C}^N$  se define su *transformada coseno discreta* (abreviada habitualmente mediante DCT) como

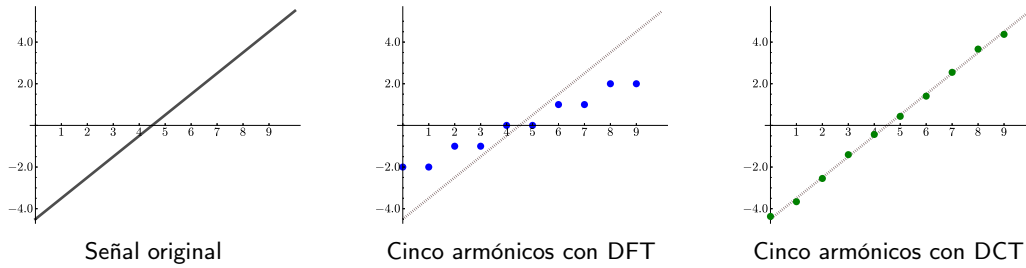
$$(22) \quad \hat{f}^c(n) = \sum_{m=0}^{N-1} f(m) \cos\left(\frac{\pi n}{N}\left(m + \frac{1}{2}\right)\right).$$

El desarrollo de Fourier correspondiente es

$$(23) \quad f(m) = \frac{\widehat{f}^c(0)}{N} + \frac{2}{N} \sum_{n=1}^{N-1} \widehat{f}^c(n) \cos\left(\frac{\pi n}{N}\left(m + \frac{1}{2}\right)\right).$$

Para ver que la idea funciona, tomemos la señal  $f(n) = n - 4.5$  en  $\mathbb{Z}_{10}$ . Los valores de la transformada coseno discreta son  $\widehat{f}^c(0)$  si  $n$  es par y  $\widehat{f}^c(1) = -20.1801$ ,  $\widehat{f}^c(3) = -2.1615$ ,  $\widehat{f}^c(5) = -0.7071$ ,  $\widehat{f}^c(7) = -0.2859$ ,  $\widehat{f}^c(9) = -0.0801$ .

Veamos gráficamente lo que ocurre al reconstruir la señal original empleando sólo la primera mitad de los armónicos:



No es ya sólo que la transformada coseno discreta (DCT) da una aproximación casi perfecta con la mitad de la información, es que la transformada de Fourier discreta apenas aproxima en la misma situación.

Para probar (23), por la linealidad, basta considerar el caso  $f(m) = \delta(m - m_0)$ , esto es  $f(m_0) = 1$  y  $f(m) = 0$  para  $m \neq m_0$ . Por sustitución directa y empleando  $2 \cos \alpha \cos \beta = \cos(\alpha + \beta) + \cos(\alpha - \beta)$ , la fórmula se reduce en este caso a

$$(24) \quad N\delta(m - m_0) = 1 + \sum_{n=1}^{N-1} \left( \cos\left(\frac{\pi n}{N}(m_0 + m + 1)\right) + \cos\left(\frac{\pi n}{N}(m - m_0)\right) \right) \quad \text{para } 0 \leq m < N.$$

La suma de cosenos es par en  $n$  y se anula para  $n = N$ , por tanto se reescribe como

$$(25) \quad 2N\delta(m - m_0) = \sum_{n=-N}^{N-1} \left( \cos\left(\frac{\pi n}{N}(m_0 + m + 1)\right) + \cos\left(\frac{\pi n}{N}(m - m_0)\right) \right) = \Re \sum_{n=0}^{2N-1} \left( e\left(\frac{n(m_0 + m + 1)}{2N}\right) + e\left(\frac{n(m - m_0)}{2N}\right) \right)$$

y esto se sigue de (2) con  $2N$  en lugar de  $N$ , notando que  $0 < m_0 + m + 1 < 2N$ .

## Aplicaciones

El análisis de Fourier es fundamental en las telecomunicaciones y otras ramas de la ingeniería. Entre la multitud de aplicaciones, señalamos aquí algunas de ellas que cumplen los requisitos de ser suficientemente sencillas y tener relación con la informática.

**El formato JPEG.** El color de un píxel (sin transparencias) se expresa mediante tres números de un *byte*, es decir, enteros entre 0 y 255, que corresponden a los canales  $R$  (rojo),  $G$  (verde) y  $B$  (azul). Cuando los valores  $R$ ,  $G$  y  $B$  coinciden, el color resultante es un tono de gris. De esta forma, una fotografía de tamaño  $N \times M$  puede considerarse como una función  $f : \{0, 1, \dots, N - 1\} \times \{0, 1, \dots, M - 1\} \rightarrow \mathbb{R}^3$  y para fotografías en blanco y negro (en rigor, en tonos de gris),  $\mathbb{R}^3$  se puede reemplazar por  $\mathbb{R}$ .

El formato JPEG es el más utilizado en la fotografía digital no profesional. Las siglas se refieren al comité que introdujo sus especificaciones (*Joint Photographic Experts Group*) pero ha quedado como nombre del propio formato.

En lo que respecta al análisis de Fourier, el formato trata imágenes de tamaño extremadamente pequeño  $8 \times 8$  pues cualquier otra se divide en bloques de este tamaño (si las dimensiones no son múltiplos de 8, se añaden filas o columnas artificialmente). Cada uno de estos bloques será una función  $\{0, 1, \dots, 7\} \times \{0, 1, \dots, 7\} \rightarrow \mathbb{R}^3$ . En este proceso de descomposición en bloques y su traducción en funciones, hay un punto técnico que complica la implementación aunque es poco relevante en el modelo matemático. El espacio de colores  $RGB$  no refleja muy bien las capacidades de nuestra visión, que es mucho más sensible a la luminosidad que a los tonos de color, especialmente a los de azul. Por ello se define el espacio de color  $Y C_B C_R$ , donde  $Y$  es la *luminancia* (o *luma*) y el par  $C_B$  y  $C_R$  es la *chrominancia* (o *croma*) que tiene menos importancia. La relación teórica entre ambos espacios de color es la siguiente [Sal02, p.145]

$$(26) \quad \begin{pmatrix} Y \\ C_B \\ C_R \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \frac{1}{256} \begin{pmatrix} 77 & 150 & 29 \\ -44 & -87 & 131 \\ 131 & -110 & -21 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix},$$

aunque en la práctica no hay que tomarla literalmente porque queremos ajustar los resultados para que sean enteros de un *byte*.

Como  $C_B$  y  $C_R$  son menos importantes, al tratar estos canales se suelen tomar sólo la mitad o la cuarta parte de los píxeles de la imagen (con GIMP, al exportar como *jpg* se puede elegir entre 4 posibilidades en **Advanced options>subsampling**). Para imágenes en tonos de gris,  $C_B$  y  $C_R$  son constantes y toda la información está en  $Y$ . en cualquier caso, los tres canales se tratan de manera matemáticamente similar, por tanto supondremos que un bloque  $8 \times 8$  está representado por una función  $F : \{0, 1, \dots, 7\} \times \{0, 1, \dots, 7\} \rightarrow \mathbb{R}$ .

Aplicando la transformada coseno discreta sucesivamente a cada una de las variables se obtiene

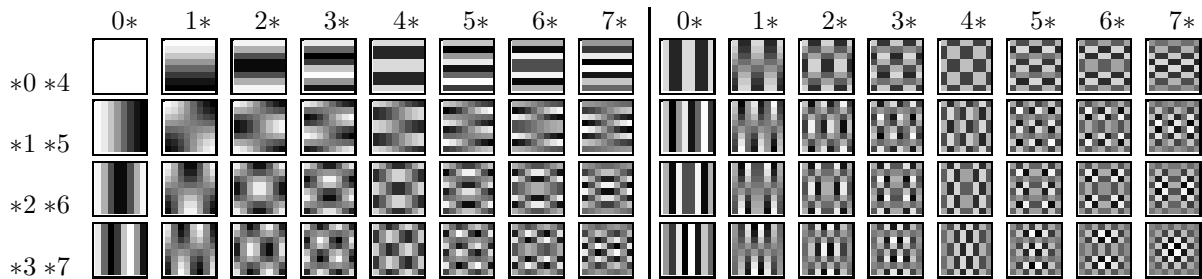
$$(27) \quad F = \sum_{n=0}^7 \sum_{m=0}^7 a_{nm} \phi_{nm} \quad \text{con} \quad \phi_{nm}(k, l) = \cos\left(\frac{\pi n}{16}(2k+1)\right) \cos\left(\frac{\pi m}{16}(2l+1)\right)$$

y los coeficientes vienen dados por

$$(28) \quad a_{nm} = \frac{\alpha_n \alpha_m}{64} \sum_{k=0}^7 \sum_{l=0}^7 F(k, l) \phi_{nm}(k, l) \quad \text{donde} \quad \alpha_n = \begin{cases} 2 & \text{si } n \neq 0 \\ 1 & \text{si } n = 0 \end{cases}$$

El objetivo que persigue es comprimir (con pérdidas) la foto omitiendo totalmente cierto número de coeficientes de Fourier  $a_{nm}$  y perdiendo parte de la precisión en el resto. Con los datos resultantes se lleva a cabo una compresión convencional. Esto explica por qué las carpetas con ficheros *.jpg* apenas sufren cambios en su tamaño cuando se les aplican programas de compresión y también por qué editar muchas veces una foto (abrir un JPEG y guardarlo como JPEG) reduce su calidad.

De alguna forma, (27) representa cualquier imagen  $8 \times 8$  como superposición de imágenes básicas  $8 \times 8$ . Asignando al negro el  $-1$  y al blanco el  $1$ , el aspecto de estas imágenes básicas es:



En la mitad izquierda están los armónicos  $\phi_{nm}$  con  $0 \leq m < 4$  y en la derecha, los armónicos  $\phi_{nm}$  con  $4 \leq m < 8$ .

Los coeficientes de Fourier  $a_{nm}$  se calculan con cierta precisión, a veces aproximando a enteros (de nuevo, con GIMP, **Advanced options>DCT method** da cierto control). Parece claro, y las experiencias lo avalan, que el



ojo es poco sensible a oscilaciones de brillo en áreas pequeñas, es decir, que ve peor los armónicos con frecuencias mayores. La *matriz de cuantificación* es una matriz entera  $C = (c_{nm})_{n,m=0}^7$  de naturaleza experimental tal que  $c_{nm}$  está en relación inversa a la “visibilidad” del armónico  $\phi_{nm}$ . La matrices más empleadas para la luminancia y la crominancia, son respectivamente:

$$(29) \quad C = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

En vez de los coeficientes de Fourier  $a_{nm}$ , lo que se almacena son los enteros  $\lambda_{nm}$  más cercanos a  $a_{nm}/c_{nm}$ .

Matemáticamente, el esquema del proceso desde  $F$  a los números que hay que almacenar es

$$(30) \quad F \rightarrow \{a_{nm}\}_{n,m=0}^8 \rightarrow \{\lambda_{nm} = [a_{nm}/c_{nm} + 1/2]\}_{n,m=0}^8,$$

donde  $[\cdot]$  indica la parte entera.

Si  $F$  viene dada por 64 valores enteros y ahora almacenamos los  $\lambda_{nm}$  que son otros 64 enteros, cabe preguntarse dónde está la compresión. La respuesta es que típicamente los  $\lambda_{nm}$  son muchas veces cero o valores pequeños, y en estas condiciones, la lista de todos los  $\lambda_{nm}$  para todos los bloques  $8 \times 8$  de la imagen original tendrá muchos ceros y valores repetidos. No sorprenderá a nadie que una tabla de datos con estas características es muy fácil de comprimir con métodos convencionales (de compresión sin pérdidas) y el fichero *jpg* que vemos en nuestro ordenador es el resultado de esa compresión. Habitualmente se emplea la *compresión de Huffman* porque es muy ventajosa cuando las probabilidades de diferentes símbolos están muy descompensadas y porque está libre de patentes.

Cuando se abre una imagen JPEG, internamente en el ordenador se descomprime el fichero y se obtienen los  $\lambda_{nm}$ , con ellos se aproximan los  $a_{nm}$  por medio de  $a_{nm} \approx c_{nm}\lambda_{nm}$ . Finalmente se utiliza (27) para reconstruir cada porción  $8 \times 8$  de la imagen.

A pesar de que el formato JPEG logra habitualmente una compresión notoria sin pérdida de calidad aparente, está claro que tiene ciertas deficiencias. La más clara es que la división en porciones  $8 \times 8$  es arbitraria (y además las porciones son demasiado pequeñas con los tamaños actuales de las fotos y monitores). Una foto puede tener zonas mucho mayores con gradaciones de color en las que el análisis de Fourier discreto funciona bien. Lo ideal sería que hubiera un proceso adaptativo en el que el tamaño de los armónicos se ajustase a las diferentes zonas de la imagen. Esta idea se manifiesta en gran medida en el nuevo formato JPEG 2000 pero, al parecer, no ha tenido el éxito que se esperaba.

**Aproximación de funciones.** Los cursos de matemáticas sobre análisis de Fourier suelen dedicar bastante tiempo a estudiar condiciones de poca regularidad en las que todavía se tiene (5) y a cambiar la definición de la convergencia para que ésta se produzca. Muchas veces se pierde de vista el hecho de que para funciones con mucha regularidad, la aproximación mediante unos pocos términos de la serie de Fourier es realmente buena.

La aplicación que perseguimos es desarrollar un método que un ordenador o una calculadora pudiera utilizar internamente para aproximar funciones trascendentes. El método que veremos es marginal en la práctica frente a otros más eficientes o precisos (*CORDIC* o *aproximantes de Padé*) pero es muy fácil de implementar. Pensemos, por ejemplo, en qué algoritmo podría usar una calculadora al pulsar la tecla *sin* o un ordenador al llamar a esta función en un lenguaje de programación. Consideremos como operaciones básicas las sumas, restas, multiplicaciones y divisiones, porque están muy cerca de lo que hace el microprocesador. Así pues, el problema es cómo reducir funciones “complicadas” a estas operaciones. Una solución ingenua es utilizar polinomios de Taylor. Esto suele ser muy poco práctico porque la aproximación de Taylor es muy buena, óptima en algún sentido, cerca del punto en el que se hace pero posiblemente bastante mala en cuanto nos alejamos un poco.

La situación típica es que necesitemos aproximar la función en un intervalo acotado. Por ejemplo, si  $f(x) = \sin x$  entonces, por las simetrías y la periodicidad, basta considerar el intervalo  $[0, \pi/2]$ , si  $f(x) = \log x$  la relación  $\log(x/2^n) = \log x - n \log 2$  y que para un ordenador es trivial dividir por  $2^n$ , reduce el rango a  $[1, 2]$ .

Con una transformación lineal podemos suponer que el intervalo en el que queremos llevar a cabo la aproximación de una función  $f$  es  $[-1, 1]$ . Definamos  $g(x) = f(\cos(2\pi x))$  que, al ser real y par, tiene un desarrollo de Fourier (5) de la forma

$$(31) \quad f(\cos(2\pi x)) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos(2\pi n x) \quad \text{con} \quad a_n = \int_0^1 g(x) \cos(2\pi n x) dx.$$

Supongamos que  $f$  es muy regular, como ocurre con casi todas las funciones que aparecen en las teclas de nuestra calculadora en los intervalos de interés. Entonces  $\{a_n\}_{n=0}^{\infty}$  tiende a cero muy rápidamente. Si cambiamos  $x$  por  $\frac{1}{2\pi} \arccos x$  en el desarrollo anterior, se tiene

$$(32) \quad f(x) = a_0 + 2 \sum_{n=1}^{\infty} a_n T_n(x) \quad \text{con} \quad T_n(x) = \cos(n \arccos x).$$

Diremos que éste es el *desarrollo de Chebyshev-Fourier* porque a los  $T_n$ ,  $n \geq 0$ , se les llama *polinomios de Chebyshev*. Quizá resulte una sorpresa que, como su nombre indica, son realmente polinomios para  $-1 \leq x \leq 1$ . Una manera rápida de comprobarlo es partir de  $T_0 = 1$  y  $T_1 = x$  y utilizar las fórmulas de adición para el seno y el coseno para deducir

$$(33) \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad \forall n \in \mathbb{Z}^+.$$

Esta recurrencia implica que evaluar estos polinomios es computacionalmente muy poco costoso. La aproximación de  $f$  se logra tomando un número finito de términos en (32). Los coeficientes  $a_n$  necesarios para efectuar la aproximación con la precisión deseada se precálculan con precisión al diseñar el *software* y se almacenan en memoria. En una calculadora formarían parte de su propia “arquitectura”.

Tomemos como ejemplo la aproximación de la función seno. Elegimos,  $f(x) = \sin(\pi x/2)$ , aunque en principio parece mejor  $\sin(\pi(x+1)/4)$  porque sus valores en  $[-1, 1]$  determinan sin redundancias los de la función seno, pero  $f$  tiene la ventaja computacional de que su simetría impar implica  $a_{2k} = 0$  (ejercicio). Si tomamos tres coeficientes de Fourier no nulos, precálculados con 8 decimales, (32) lleva a

$$(34) \quad \sin\left(\frac{\pi}{2}x\right) \approx 0.56682409T_1(x) - 0.06903589T_3(x) + 0.00224536T_5(x),$$

donde  $T_1(x) = x$ ,  $T_3(x) = 4x^3 - 3x$  y  $T_5(x) = 16x^5 - 20x^3 + 5x$ , aunque las expresiones explícitas para los  $T_n(x)$  son indiferentes desde el punto de vista computacional porque tenemos la recurrencia (33). Se puede comprobar que el error absoluto máximo en esta aproximación es menor que  $7 \cdot 10^{-5}$ . Tomando 8 coeficientes el error máximo está ya cercano al habitual en *doble precisión*,  $2^{-53} \approx 10^{-16}$ . Es decir, que almacenando sólo 8 números conseguimos tanta aproximación para el seno de cualquier número real como la que nos da nuestro ordenador.

**Filtros de convolución.** Supongamos una señal  $f : \mathbb{Z}_N \rightarrow \mathbb{C}$  con  $N$  grande, digamos que idealmente tenemos  $f : \mathbb{Z} \rightarrow \mathbb{C}$ , que de algún modo equivaldría a  $N = \infty$ . La convolución con  $g : \mathbb{Z} \rightarrow \mathbb{C}$  corresponde a efectuar la suma  $\sum f(n-m)g(m)$  y es natural, por razones prácticas, limitarse a funciones  $g$  con soporte acotado. En ingeniería este tipo de filtros de convolución se dice que son *filtros de respuesta finita*, abreviado como FIR (finite impulse response). Muchas veces se representa  $g$  con un vector de cuyas coordenadas son los valores consecutivos de  $g$  forzando a que  $g(0)$  sea aproximadamente la coordenada en la posición intermedia. Por ejemplo,  $(1, 2, 3, 2, 1)$  indica la función  $g(n) = 3 - |n|$  para  $|n| \leq 2$  y cero en el resto.

Como ya hemos apuntado,  $f \mapsto f * g$  lleva a cabo una especie de promedio ponderado de  $f$  a través de  $g$  que dentro del análisis de Fourier es equivalente a la multiplicación de los coeficientes de Fourier y por tanto sirve para aumentar o reducir la importancia de las diferentes componentes de frecuencia. Sin embargo, no es

necesario saber nada de análisis de Fourier para entender el funcionamiento y el interés de algunos filtros de convolución básicos que incorporan los programas de retoque fotográfico. Lo que sigue es totalmente elemental pero aún así interesante desde el punto de vista práctico.

Ya sabemos, que una fijado uno de los canales del espacio de color, una imagen  $N \times M$  se puede considerar como una función  $f : \{0, 1, \dots, N - 1\} \times \{0, 1, \dots, M - 1\} \rightarrow \mathbb{R}$ . Para la teoría, supongamos inicialmente que  $f : \mathbb{Z}^2 \rightarrow \mathbb{R}$  (imagen de dimensiones infinitas). En analogía con lo anterior, la convolución se define como  $(f * g)(n, m) = \sum \sum f(n - i, m - j)g(i, j)$ . Si  $g$  tiene soporte incluido en  $[-K, K]^2$ , los números  $g(i, j)$  pueden interpretarse como elementos de una matriz cuadrada  $C$  de dimensión  $2K + 1$ . Dada una matriz de este tipo, el filtro de convolución actúa sobre la imagen como

$$(35) \quad \text{Nuevo valor del píxel } (n, m) = \sum_{i=-K}^K \sum_{j=-K}^K f(n - i, m - j)c_{i+K+1, j+K+1}.$$

Esto no es más que cambiar el valor de cada píxel por un promedio de los valores de píxeles de alrededor, ponderado a través de  $C$ .

Por ejemplo, consideremos las matrices,

$$(36) \quad C_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad C_2 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{y} \quad C_3 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}.$$

La primera matriz da lugar a un filtro trivial que no tiene ninguna acción, la segunda, promedia entre los píxeles adyacentes, con lo cual se producirá cierto desenfoque. Dicho sea de paso, el desenfoque más natural desde el punto de vista teórico consiste en tomar una matrices que aproxime los valores de una función gaussiana  $K_1 \exp(-K_2(x^2 + y^2))$  y es lo que aproxima el filtro de *desenfoque gaussiano* que está en los programas de retoque fotográfico. La última matriz, dejará fijo el valor de un píxel si los adyacentes tienen el mismo valor y lo cambia mucho si hay grandes variaciones. Este tipo de filtros se utiliza para detectar bordes en imágenes.

En la práctica, la fórmula (35) no se aplica tal cual. Hay dos problemas a tratar, el primero es que la imagen  $f$  no es ilimitada y hay que decidir qué se hace en los bordes, habitualmente uno inventa una regla para continuar la imagen con los valores que sean necesarios (por ejemplo repetir las últimas  $K$  filas y columnas). El segundo es que, sin propiedades demasiado restrictivas sobre  $C$ , en general el valor de un píxel puede tomar valores no enteros o fuera del rango  $[0, 255]$ , una solución fácil y común es aproximar al entero más próximo y, si fuera necesario, asociar los que caen en  $(-\infty, 0)$  y  $(255, \infty)$  a 0 y 255, respectivamente (este proceso se conoce en inglés con el nombre de *clamping*).



Imagen original

Filtro  $C_3$  (negativo)Filtro  $C'_3$  (negativo) $\frac{1}{8}C'_3$  (negativo)  $[0, 240]$ 

Veamos un ejemplo de estas ideas intentando diseñar a partir de  $C_3$  un filtro que simule la transformación de una foto en un dibujo hecho a lápiz. Tomamos una imagen en tonos de gris (la de la izquierda) y aplicamos  $C_3$

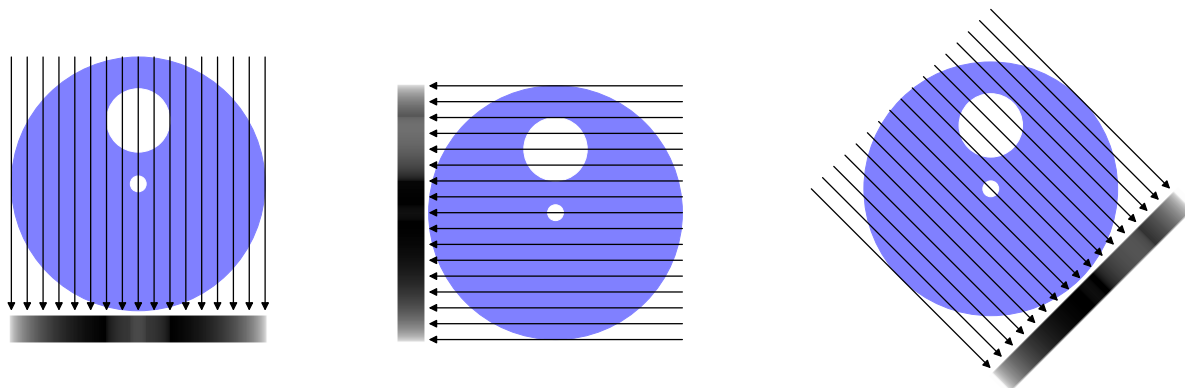
efectuando el *clamping* como se ha indicado y mostramos en negativo la imagen (el 0 será blanco y el 255 negro) para que las partes de “lápiz” sean oscuras. El resultado es la segunda figura. y parece poco denso y aleatorio porque el filtro  $C_3$  detecta variaciones en un rango de 3 pixels, que es una unidad pequeña (en un monitor actual típicamente  $0.75mm$  y todavía menor en los dispositivos móviles de calidad).

Esto sugiere trabajar en bloques de tamaño doble. Es decir, consideramos  $C'_3$  como  $C_3$  pero sustituyendo cada elemento por una celda  $2 \times 2$  con todos sus valores iguales. El resultado es la tercera figura. Simula mejor un trazado a lápiz, pero tiene demasiadas partes oscuras. La razón es que, como  $C'_3$  involucra más números no nulos, tras aplicar el filtro, hay muchos valores que sobrepasan 255 (negro en el negativo). Para evitarlo, en la última figura se ha dividido el filtro por 8 y se ha bajado el umbral de negro hasta 240 para mejorar el resultado.

Casi todo los filtros de los programas de retoque fotográfico son combinaciones de convoluciones con otras ideas sencillas. En el conocido programa libre GIMP, es posible elegir una matriz de convolución para crear un filtro propio (en el menú `Filters>Generic>Convolution Matrix`). En la documentación correspondiente se lee “*Here is a mathematician’s domain. [...] What is a convolution matrix? It’s possible to get a rough idea of it without using mathematical tools that only a few ones know.*” Lo curioso es que a los matemáticos en formación que conocen sobradamente las herramientas, rara vez se les explica su utilidad.

**Tomografías y transformada de Radon.** Una última aplicación está en la base de la obtención de algunas imágenes médicas.

La TAC (tomografía axial computarizada) es una técnica poco invasiva para obtener imágenes de la estructura de secciones planas de tejidos sin necesidad de cirugía. Esencialmente lo que se hace es tomar muchas radiografías unidimensionales en muchas direcciones contenidas en el plano que quiere examinarse. Lo que se hace es reconstruir la sección a partir de sus “sombras”, teniendo en cuenta que el tejido es semitranslúcido a los rayos X.



Para formular el modelo, consideremos un fino haz de rayos X que atraviesa la muestra siguiendo una recta  $r_{\theta,t}$  que dista  $t$  del origen y cuya normal forma un ángulo  $\theta$  con el eje  $OX$ . Es natural suponer que la atenuación que sufre al atravesar la muestra depende de la integral densidades que atraviesa, es decir, de

$$(37) \quad P_{\theta}(t) = \int_{r_{\theta,t}} \rho \quad \text{con} \quad r_{\theta,t} \equiv x \cos \theta + y \sin \theta = t.$$

Matemáticamente, esto es lo que se llama *transformada de Radon* (a veces con una normalización diferente) o, por razones obvias, *transformada de rayos X*.

El problema que se plantea es cómo recuperar  $\rho$  a partir de  $P_{\theta}(t)$ , que idealmente se conocería con infinitas radiografías. Usando la definición de integral de línea de una función escalar con la parametrización de  $r_{\theta,t}$  dada por  $x = t \cos \theta - u \sin \theta$ ,  $y = t \sin \theta + u \cos \theta$ , se tiene

$$(38) \quad P_{\theta}(t) = \int_{-\infty}^{\infty} \rho(t \cos \theta - u \sin \theta, t \sin \theta + u \cos \theta) du$$

Tomando transformada de Fourier (para  $\theta$  fijo),

$$(39) \quad \widehat{P}_\theta(r) = \int_{\mathbb{R}^2} \rho(t \cos \theta - u \sin \theta, t \sin \theta + u \cos \theta) e(-tr) \, dudt = \int_{\mathbb{R}^2} \rho(x, y) e(-rx \cos \theta - ry \sin \theta) \, dxdy$$

donde se ha hecho el cambio de variable  $(t, u) \mapsto (x, y)$  que corresponde a la parametrización anterior.

Para seguir, tenemos que introducir la transformada de Fourier en varias variables. Es inmediato generalizar (8) a dimensiones superiores, simplemente aplicando la misma fórmula en cada variable. Así para dimensión dos

$$(40) \quad f(x, y) = \int_{\mathbb{R}^2} \widehat{f}(\xi, \eta) e(\xi x + \eta y) \, d\xi d\eta \quad \text{donde} \quad \widehat{f}(\xi, \eta) = \int_{\mathbb{R}^2} f(x, y) e(-\xi x - \eta y) \, dxdy.$$

De esta forma, la conclusión de los cálculos anteriores es

$$(41) \quad \widehat{P}_\theta(r) = \widehat{\rho}(r \cos \theta, r \sin \theta)$$

y la primera fórmula de (40) permite recuperar  $\rho$ , como deseábamos a partir de esta información. Haciendo un cambio a polares, el resultado final es

$$(42) \quad \rho(x, y) = \int_0^{2\pi} \int_0^\infty r \widehat{P}_\theta(r) e(xr \cos \theta + yr \sin \theta) \, dr d\theta.$$

En la práctica,  $\rho$  no tendrá mucha regularidad y entonces la aproximación de la transformada de Fourier y de las integrales no es sencilla. Quizá esto explique que desde los resultados matemáticos de J. Radon hasta que las tomografías formaron parte de la práctica médica habitual, pasaron más de 70 años. Por otro lado, el método indicado no es el único con el que se lleva a cabo la reconstrucción. Lo que está claro es que para el diagnóstico médico actual es realmente crucial disponer de técnicas poco invasivas como ésta. Imágenes mucho más precisas se obtienen con la *resonancia magnética nuclear* (cuando es posible llevarla a cabo). Sus bases físicas son mucho más complicadas pero hay también una relación con los fenómenos oscilatorios: se busca que entren en resonancia una onda asociada a los núcleos de átomos de hidrógeno y una onda electromagnética externa.

**Algunos comentarios sobre formatos de audio.** Ciertamente el que dijo aquello de que “una imagen vale más que mil palabras” no tenía en mente el mundo digital. Una foto  $3648 \times 2736$  tomada con una cámara digital ocupa en formato JPEG, alrededor de un par de megas, teniendo una calidad más que suficiente para un fondo de escritorio o impresión en formato medio. Por otro lado, con parámetros habituales, la amplitud de un sonido en estéreo se representa con 32 bits (16 por cada canal) y se muestrea a 44.1kHz, es decir, que 44100 veces por segundo, hay que procesar un número de 32 bits esto es alrededor de 1.5 megabits por segundo. Es decir, sin ninguna compresión, 1000 palabras, que en promedio llevan unos 5 minutos, requerirían mas de 50 megas. Esto está muy por encima de lo que necesita cualquiera de las fotos que almacenamos en nuestro ordenador. Incluso si no llevamos acabo ninguna compresión (ni sin pérdidas ni con pérdidas), una imagen  $3648 \times 2736$  sólo podría llegar a ocupar 29 megas.

Los números anteriores indican que no es conveniente, ni a veces posible, almacenar audio sin una severa compresión. En los años 90 se creó el formato MP3 que con el posterior auge de la transmisión por internet de ficheros multimedia se convirtieron en el gran referente en audio (Ogg vorbis es un formato libre análogo pero ha tenido menos difusión).

Un CD conteniendo audio sin comprimir tiene una duración de 74 minutos, mientras que el mismo contenido en MP3 típicamente multiplica esta duración por más de 10. Por supuesto, el precio que hay que pagar es una pérdida de calidad. La difusión del formato no deja dudas acerca de que esta pérdida es aceptable por la mayor parte de los usuarios.

Desafortunadamente, la descripción completa del formato es muy compleja y no está demasiado divulgada (véase [Sal02] para una descripción somera). Esta complejidad se manifiesta especialmente en la parte de la codificación. La fuerza del algoritmo está en un modelo psicoacústico (basado en experimentos) para filtrar la señal. el sonido se divide inicialmente en subbandas de frecuencias (que podemos interpretar como fragmentos de la serie de Fourier). La anchura de las subbandas y la imprecisión en su tratamiento es mayor cuanto menor sea la sensibilidad del oído humano a ellas. También hay un sistema adaptativo de ventanas que permite considerar ondas regularizadas más largas si no ha habido cambios significativos.

Menos evidente pero importante, es aprovechar el *enmascaramiento* (*masking*) temporal o tonal en el filtrado de la señal. Éste es un fenómeno relacionado con la fisiología de nuestra audición que consiste en que cuando la amplitud se eleva súbitamente, nuestro oído apenas percibe la señal (se queda “sordo”) en un instante posterior e incluso ligeramente anterior. De la misma forma, una frecuencia dominante causa una insensibilidad a frecuencias cercanas que se pueden filtrar sin consecuencias.

## Referencias

- [CT65] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.
- [Fou88] J. Fourier. *Théorie analytique de la chaleur*. Éditions Jacques Gabay, Paris, 1988. Reprint of the 1822 original.
- [Ger99] N. Gershenfeld. *The nature of mathematical modeling*. Cambridge University Press, Cambridge, 1999.
- [HJB85] M. T. Heideman, D. H. Johnson, and C. S. Burrus. Gauss and the history of the fast Fourier transform. *Arch. Hist. Exact Sci.*, 34(3):265–277, 1985.
- [Kat76] Y. Katznelson. *An introduction to harmonic analysis*. Dover Publications, Inc., New York, corrected edition, 1976.
- [Sal02] D. Salomon. *A guide to data compression methods*. Springer-Verlag, 2002.
- [Str93] G. Strang. Wavelet transforms versus Fourier transforms. *Bull. Amer. Math. Soc. (N.S.)*, 28(2):288–305, 1993.